Abstract—This paper argues the importance of building an early warning system to improve the academic performance of post-secondary science, technology, engineering, and mathematics (STEM) education. It emphasizes the necessity of identifying not only the students who are at the risk of failing or dropping out of the course, but also the students who are not performing good (obtaining grade C) as well as who are doing good (grade A or B). For increasing the likelihood of graduation and improving retention, it is imperative to send early signals about predicted performance to the entire spectrum of the class. To accomplish this goal, this paper leverages Machine Learning based predictive models. More specifically, it develops multi-class classifiers to identify students belonging to three groups: Prone to risk (grade C or below), OK (grade B), and Good (grade A). Through an extensive analysis, it identifies the most effective classifiers for early forecasting of student performance.

Index Terms—Predictive model, Machine Learning, Classifier, STEM Education, Early Warning

I. INTRODUCTION

While the number of new jobs that require science, technology, engineering, and mathematics (STEM) knowledge and skills is increasing in the United States of America (USA), the attrition rate in post-secondary STEM fields is high [1]–[3]. The 2013 report from the National Center for Education Statistics indicates that more than half of STEM major freshmen left these fields and more than half of STEM bachelor’s degree recipients switched to non-STEM fields at the graduate level [1], [4]. One of the factors found in this report that caused the high attrition rate is students’ poor academic performance [5]. In particular, it has been observed that students’ performance in the first few years of college plays an important role in retaining them into subsequent years [5]–[8]. Therefore, the problem of poor academic achievement during the initial years of undergraduate STEM education needs to be addressed by the educators and policy makers for increasing the retention and thereby increasing the number of STEM degree students [9], [10].

For achieving a sustained solution to this problem, a large-scale systemic change is proposed [2], [11]. However, implementing this change is a slow process as it needs to be tailored according to an institution’s STEM population requirements and involves a significant fiscal investment [10]. Thus, there is an imminent need to develop an effective yet fast solution that involves little systemic change. There has been a recent development of building early warning systems to improve student performance [7], [12]–[14]. These early warning systems use students’ performance data to create Machine Learning (ML) based predictive models. The predictive models are mainly used to identify at-risk students in large classroom setting [12]. While developing an early warning system for at-risk students is an important first step to improve academic performance, it is equally important to provide timely warning to the students who are not at-risk yet on the verge of risk, who are performing poorly as well as to the students who are doing good in the course. This paper builds a ML based early warning system for a university level STEM course to provide warning signals to students who are prone to risk (risk of getting letter grade C, D or F) as well as to the students who are doing okay (letter grade B) and good (letter grade A). This system will alert “Prone-to-risk” and “OK” students from the beginning of the semester to take precautions and inform “Good” students to motivate them to stay on track.

A. Research Problem

Previously predictive models were developed to identify both at-risk and not-at-risk students [12]. Students who obtain a grade lower than C (i.e., D, W or F) are labeled as “at-risk”. While it is very important to provide early warning signals to this group of students, often time at-risk students are a small subset of the student population (usually less than 10% [12]). To improve the STEM academic performance it is imperative to send timely signals not just to the students who are at risk, but also to other student groups. Previously it has been shown that students who earn grade C are less likely to graduate [15]. For a sustainable STEM training and increasing retention it is important to obtain grades better than C. This can be accomplished by informing students about their projected academic standing at a fine grained level.

In this paper, the student performance is categorized into three groups: Prone-to-risk (grade C or below), OK (grade B) and Good (grade A). The “Prone-To-Risk” group includes students who are at the risk of failing, withdrawing as well as obtaining C or D grade. Unlike the previous approaches [12] that don’t include students obtaining C grade in the at-risk group, this group of students are considered to be prone to risk.
Unless students who could obtain C are warned about their performance they might eventually degenerate and therefore needs early warning. Students with a projected grade B belong to the “OK” group. This group needs warning to improve their performance to elevate to the “Good” students group whose projected grade is A. The “Good” students’ group also needs a motivating signal to be informed of their current standing to keep up their good work.

B. Research Hypotheses and Goal

Our goal is to develop a three-class classification model for early identification of three groups of students. We plan to use this model to generate four predictions throughout the semester. Previous research suggests that there is no single model that can perform well across different classes [12]. Adding to this observation, our initial experimentation indicates that even the models could vary across the predictions. Thus, we intend to (i) find the models that are most effective identifying a specific class as well as (ii) find the models that are effective across four predictions.

For experimentation, we choose a list of classification models from previous work [12]. Since our classification task is more involved and our models are expected to identify more intricate pattern, we enrich our list of models by including three types of ML models: weak learners, strong learners and ensemble learners. We hypothesize that with less features during initial predictions, weak learners such as Naive Bayes Classifier or Logistic Regression would perform well. With the increase in the number of features strong learners such as Multi-Layer Perceptron would be more effective. Finally, with the inclusion of many features as data displays non-trivial pattern we would need to use ensemble techniques such as bagging and boosting [16].

While experimenting with the list of classifiers to build the predictive models the following questions are addressed.

1) What are the most effective ML models (models that perform most accurate classification) for performing multi-class classification based on course performance data?
2) Do the effective models vary across multiple predictions throughout the semester and why?
3) What type of model perform better during initial predictions and final predictions?
4) Can a unique model be found to identify three classes as effectively? Or do we need to find different models for identifying different classes?

The remainder of this paper is organized as following. First, the models are presented in Section II including description of the training and tuning of the models. Then, an extensive comparative analysis of the models is provided in Section III. Finally, the paper is concluded with a summary of our observations and discussion of future work in Section IV.

II. METHODOLOGY

The predictive models are built based on the data obtained from a sophomore level STEM core course on the Computer Science and Engineering major at a large Midwestern university. The final grading evaluation is based on weekly quizzes, homework assignments, midterm and final exams. This type of grading is common in the US universities. The performance data is collected from 538 students who were enrolled in this course from Fall 2015 to Fall 2019. The performance data for Spring 2019 is not used because of the lack of consistency in the number of quizzes. During Fall semester typical enrollment is more than 100 students while during Spring semester it is little over 50.

A. Dataset

In all experiments, the data are split into training and testing sets using 80% and 20% split, respectively. In other words, out of total 538 students, data obtained from 430 students is used in training the models and 108 for testing. The data are labelled with different performance levels according to the following criteria: Good ≥ 90%, 80% ≤ OK < 90%, and Prone-To-Risk < 80%. Unlike previous approaches [12] four predictions are generated throughout the semester.

B. Features

The course performance data are used as the features. Previous research shows that course performance data could be used as effective features for building predictive models [12].

To select the features, we perform Exploratory Data Analysis and compute the feature correlation with the final grade. Features with significant correlation (more than 0.45) are used for training the models. Table I provides the features used for generating four predictions.

| TABLE I | THE DURATION OF EACH PREDICTION AND THE FEATURES USED IN EACH ONE. |
|---------------------------------|---------------------------------|-----------------|
| First prediction                | Quiz 1 & Homework 1             | Week 1 – 3      |
| Second prediction               | Quiz 1 – 3 & Homework 1, 2      | Week 1 – 6      |
| Third prediction                | Quiz 1 – 5 & Homework 1, 2, 3 & Midterm 1 | Week 1 – 9   |
| Fourth prediction               | Quiz 1 – 7 & Homework 1, 2, 3, 4 & Midterm 1 | Week 1 – 12   |

C. Number of Predictions

We generate four predictions to enable the course instructor to learn more about each student as well as it is more beneficial for the students. The first prediction gives students a lead regarding their level in class before the deadline of dropping classes. The second prediction helps students to be alert before midterm comes. The third prediction enables students to see how her grade in the midterm might influence her final grade. The fourth and the last prediction is to motivate students to prepare well for the final exam.

D. Predicting Various Performance Levels

We build predictive models to perform multi-class classification. Unlike previous works [12] that only identifies students who are at-risk and not-at-risk, in this paper students are viewed to belong to one of the three groups: Good, OK, and Prone-To-Risk. Table II provides the grouping/labeling criterion.

<table>
<thead>
<tr>
<th>Class</th>
<th>Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good</td>
<td>grade ≥ 90%</td>
</tr>
<tr>
<td>OK</td>
<td>80% &lt; grade &lt; 90%</td>
</tr>
<tr>
<td>Prone-to-Risk</td>
<td>grade &lt; 80%</td>
</tr>
</tbody>
</table>

Models that identify at-risk students (D, W or F) are beneficial for only a small subsection of the class. Our approach intends to help students of all levels. For example, students who score 90% or above can be benefited by a signal from our model so that they are motivated to keep up their good work. Similarly, students in the OK level can put more effort to belong to the Good group. The Prone-To-Risk group is made broad (less than 80%) to include students who could potentially obtain a grade C, D or F. Our goal is to help students to achieve grades above C, which is a key requirement to be successful in the advanced STEM courses and in the degree program. Thus, students belonging to this group need to put significant effort to improve their class standing.

E. Prediction Models

Seven classification techniques are used for building our predictive models. Some models, such as Logistic Regression, K-Nearest Neighbor, Naive Bayes Classifier and Multi-Layer Perceptron are used in previous work [12]. Some of these models have high bias (e.g., Logistic Regression, K-Nearest Neighbor, Naive Bayes Classifier) and some have high variance (e.g., Multi-Layer Perceptron) [16]. Models with high bias tend to underfit the data (unable to learn the pattern), while models with high variance tend to overfit (don’t generalize to unseen data). Our experimental analysis shows that with the increase in the number of features (towards the final prediction) we need to use ensemble models that are able to reduce both the bias and variance. Thus, we use ensemble techniques such as Bagging and Boosting [16]; and experiment with Random Forest Classifier, Gradient Boosting Classifier and Adaptive Boosting Classifier.

- Logistic Regression classifier (Log Reg) is the first model to be used. It is a statistical method used for analyzing the training data with one or more independent variables that determine an outcome. Logistic Regression is a reliable prediction method commonly used in educational settings [17] [18] [19]. It calculates the probability of a categorical variable (e.g., letter grade, pass/no-pass) from a number of predicting variables [20].
- K-Nearest Neighbor (KNN) is a non-parametric classification method. KNN classifies an object by a majority vote of its K neighbors [21]. In this case, the object will be the student. KNN is based on calculating the Euclidean method between two points to find the nearest neighbor. In this paper, different numbers of nearest neighbors were used depending on the prediction to identify the Prone-to-risk students.
- Naive Bayes Classifier (NBC) is a probabilistic classifier. It calculates a conditional probability distribution over the output of a function based on applying Bayes’ theorem with the (naive) assumption of independence between the predictive variables [22]. NBC has shown a great potential in identifying the Prone-to-risk students [12].
- Multi-Layer Perceptron (MLP) is a model which tries to simulate the brain using artificial neural network. This artificial neural network consists of nodes (neurons). These nodes are connected together with different weights. MLP consists of several hidden layers that are connected to the input layer. Depending on the type of network, there may be one or more outputs [23]. The number of used hidden layers differs in each prediction as the number of features changes.
- Random Forest (Rand For) is an ensemble method that can be used for either classification or regression problems. In this case, it is used as a classification learning method. This method learns by constructing multitude of decision trees. Random forest are very useful as they correct the over-fitting caused by the decision trees [24].
- Gradient Boosting Classifier (GBC) is a model in the form of an ensemble of weak prediction models, typically decision trees. Decision Tree (DT) is a modeling method based on partitioning. In each step, it partitions the data based on one variable (e.g., midterm exam grade) until all data in each node have only one category label (e.g., pass or fail) or all variables have been used [23]. It builds the model in a stage-wise fashion like other boosting methods do. The origin of boosting is learning theory and Ada-boost. Ada-Boost is most well-known boosting algorithm [25].
- Adaptive Boosting Classifier (ABC) is an iterative ensemble method. Ada-Boost classifier builds a strong classifier by combining multiple poorly performing classifiers so that you will get high accuracy strong classifier. The basic concept behind Ada-boost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations.

F. Model Evaluation Criteria

While our goal is to identify three groups correctly, we emphasize identifying the Prone-To-Risk (PR) group. Thus, we intend to achieve higher recall for this group (Eq.1). Also, we aim to achieve high predictive accuracy for all classes. Therefore, the overall accuracy is calculated (Eq.2). In addition to this, we report the F1 score (Eq.3). In order to calculate the F1 score, the precision and the recall of each model are calculated in (Eq.4) (Eq.5).
Recall (group x) = \frac{\text{True Positive (group x)}}{\text{True Positive (group x)} + \text{False Negative (group x)}} \tag{1}

Overall Accuracy = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total number of students}} \tag{2}

F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}

Overall Recall = \frac{\text{True Positives}}{\text{True Positives} + \text{True Negatives}} \tag{4}

Overall Precision = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{5}

where:

**True Positive**: number of students who are prone-to-risk and are identified as prone-to-risk.

**True Negative**: number of students who are not prone-to-risk and are not identified as prone-to-risk.

**False Negative**: number of students who are prone-to-risk but are not identified by the models as prone-to-risk.

**False Positive**: number of students who are not prone-to-risk but are identified by the models as prone-to-risk.

### III. Experimentation and Results

Seven models are trained to generate four predictions. These models perform multi-class classification (Good, OK and Prone-To-Risk). The models are optimized via hyper-parameter tuning. Finally, the performance of the models are evaluated based on the evaluation criteria in section II-F. Our goal is to identify the best performing models for the four predictions.

In a multi-class classification each training point belongs to one of \( N \) different classes. The goal is to construct a function which, given a new data point, will correctly predict the class to which the new point belongs. In this case, three different classes are used: Good, OK, and Prone-To-Risk.

A confusion matrix is used to present the evaluation of the models using test data. A confusion matrix is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. Figure 1 shows an example of the confusion matrix. Each student can fall under nine groups. These groups are: True Positive Good (TPG), False Negative Good (FNG), False Positive Good (FPG), True Positive OK (TPK), False Negative OK (FNK), False Positive OK (FPK), True Positive Prone-To-Risk (TPPR), False Negative Prone-To-Risk (FNPR) and False Positive Prone-To-Risk (FPFR).

The matrix in figure 1 shows how many students fall in each group. For example, there are 42 students that are grouped as TPG by the example model. In other words, the model in the example chose 42 students to be good student and they are actually good students according to their final grades. On the other hand, the same model picked 20 additional students as good students, But, they are not actually good students.

### III. Experimentation and Results

In what follows, the performance analysis of the four models is presented.

#### 1) Prediction 1: the features used are only Quiz 1 and Homework 1. Due to lack of more than two features we do not perform feature selection.

The hyper-parameters are tuned for each model using the grid search method. Table III shows the hyper-parameter values used for the top three models in this prediction.

#### Table III: Hyper-parameters values for top three models in prediction 1

<table>
<thead>
<tr>
<th>Model</th>
<th>Hyper-parameter values</th>
</tr>
</thead>
<tbody>
<tr>
<td>NBC</td>
<td>N/A</td>
</tr>
<tr>
<td>Log Reg</td>
<td>multi-class = multinomial, C = 0.007, max-iter = 100, solver = newton-cg</td>
</tr>
<tr>
<td>MLP</td>
<td>random state = 42, hidden-layer-sizes = (10, 20), alpha = 0.01, solver = lbfgs, max-iter = 600, learning-rate = adaptive</td>
</tr>
</tbody>
</table>

Evaluation of the seven models for prediction 1 (using the performance data until week 3) is reported in table IV. We observe that NBC demonstrates the lowest recall for Prone-To-Risk. In other words, it has the lowest number of false negative Prone-To-Risk students. Its recall for Prone-To-Risk group is 72%, however, its total accuracy is 59.3%.

The second-best model in determining the Prone-To-Risk students is Logistic Regression. Its recall (Prone-To-Risk) is 64%, however, its total accuracy is 62.04%. The total accuracy for logistic regression is better than NBC’s. The third best performing model is MLP with 60% recall for Prone-To-Risk and 60.19% total accuracy.
The percentages mentioned in Table IV are calculated based on the number of false negatives and false positives in each model. Figure 2 shows the number of misidentified students for each model. Each graph represents the false positives and the false negatives for each class in the model. It is important to identify the number of the students in each class that are misidentified. It shows which class has most misidentifications.

2) Prediction 2: the features used are Homework 1, Homework 2, Quiz 2, Quiz 3. The feature Quiz 1 is removed due to weak correlation with final grade. The optimal hyper-parameter values for the top three models in this prediction are reported in Table V.

3) Prediction 3: the features used are Homework 1, Homework 2, Homework 3, Quiz 2, Quiz 3, Quiz 4, and Quiz 5. The feature "Quiz 1" is removed due to weak correlation with final grade. The optimal hyper-parameter values for the top three models (MLP, KNN, and Log reg) for this prediction are reported in Table VII.
The second-best model is KNN which has 88% recall for Prone-To-Risk. It means that the increase in the recall (Prone-To-Risk) in KNN from the previous prediction is 24% which is similar to the change in MLP.

Logistic Regression is the third best performing model with 84% recall for Prone-To-Risk and 81.48% for total accuracy. The total accuracy of KNN is similar to that of Logistic Regression in prediction 2. It means that both of these models’ total accuracy increased at the same rate which is 8.33%. The increase in recall for Prone-To-Risk for Logistic Regression is 12% which is less than that of KNN.

<table>
<thead>
<tr>
<th>Model</th>
<th>Hyper-parameter values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rand For</td>
<td>n-estimators = 1000, max_depth = 9</td>
</tr>
<tr>
<td>KNN</td>
<td>n-neighbors = 23, weights = uniform, p = 1</td>
</tr>
<tr>
<td>Log Reg</td>
<td>multi-class = multinomial, C = 0.007, max_iter = 500, solver = lbfgs</td>
</tr>
</tbody>
</table>

4) **Prediction 4**: the features used are Homework 1, Homework 2, Homework 3, Homework 4, Quiz 2, Quiz 3, Quiz 4, Quiz 5, Quiz 6, Quiz 7 and midterm. The feature Quiz 1 is removed due to its weak correlation with final grade. The optimal hyper-parameter values for the top three models in this prediction are reported in table IX.

<table>
<thead>
<tr>
<th>Method</th>
<th>Log Reg</th>
<th>KNN</th>
<th>Rand For</th>
<th>MLP</th>
<th>GBC</th>
<th>ABC</th>
<th>NBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>FI</td>
<td>0.815</td>
<td>0.815</td>
<td>0.766</td>
<td>0.823</td>
<td>0.730</td>
<td>0.706</td>
<td>0.793</td>
</tr>
<tr>
<td>Total Acc.</td>
<td>0.815</td>
<td>0.815</td>
<td>0.769</td>
<td>0.824</td>
<td>0.722</td>
<td>0.704</td>
<td>0.787</td>
</tr>
<tr>
<td>Total Recall</td>
<td>0.815</td>
<td>0.815</td>
<td>0.769</td>
<td>0.824</td>
<td>0.722</td>
<td>0.704</td>
<td>0.787</td>
</tr>
<tr>
<td>Recall Good</td>
<td>0.870</td>
<td>0.870</td>
<td>0.830</td>
<td>0.891</td>
<td>0.826</td>
<td>0.783</td>
<td>0.785</td>
</tr>
<tr>
<td>Recall OK</td>
<td>0.750</td>
<td>0.730</td>
<td>0.622</td>
<td>0.767</td>
<td>0.768</td>
<td>0.649</td>
<td>0.764</td>
</tr>
<tr>
<td>Recall PR</td>
<td>0.840</td>
<td>0.880</td>
<td>0.800</td>
<td>0.920</td>
<td>0.760</td>
<td>0.640</td>
<td>0.800</td>
</tr>
</tbody>
</table>

**A. Discussion**

Figure 3 presents the summary of the evaluation of seven models for four predictions. The number false negative for Prone-To-Risk group is used for comparison, which identifies the students who are actually Prone-To-Risk, but the model missed them. The results obtained from our experimentation with seven classifier models belonging to three type of learners, i.e., weak learner (NBC, Logistic Regression, KNN), strong learner (MLP) and ensemble learner (Random Forest, GBC, ABC) validate our hypotheses in section I-B. More specifically, we observe that during the first two predictions when we use relatively small number of features, the weak learners (NBC and Logistic Regression) perform the best. During third prediction with an increasing number of features, we need a strong model to extract features for revealing non-trivial pattern. Finally, for the final prediction when data experiences maximum complexity, strong learners are unable to generalize due to their high variance (e.g., MLP has the largest number of misidentifications) and weak learners underfit due to high-bias. The best performing model for this prediction is the Random Forest which is created by using the bagging ensemble technique. However, contrary to our expectation, the boosting ensemble technique did not perform well.

**IV. Conclusion and Future Work**

In this paper, we build an early warning system for improving the academic performance of the post-secondary science, technology, engineering, and mathematics (STEM) education. Unlike previous approaches that focus on identifying the
student group who are at the risk of obtaining grades D, F or W [12], our approach emphasizes the need to find student groups who are prone to risk (who could obtain grades C or below). We argue that to increase the likelihood of graduation and improve retention in STEM discipline, students need to obtain grades better than C. Thus instead of binary categorization of student groups (at-risk and not-at-risk), we intend to categorize students belonging to three groups: Prone to risk (grade C or below), OK (grade B), and Good (grade A). We leverage Machine Learning classification models for performing this categorization. Our early warning system makes four predictions throughout the semester. For improving the accuracy of the predictions we experiment with three types of ML models: weak learner, strong learner, and ensemble learner. Through an extensive optimization we identify the most effective classifiers for early forecasting of student performance.

A. Future Work

In the current work, ML models vary across the predictions. In future, we plan to develop a generalized framework for building a single predictive model. One possible approach to build such a generalized framework is to apply ensemble technique. Another line of future work could be to explore the bagging and boosting optimization techniques to improve the accuracy of the predictions with more features. In addition to this, we would like to investigate whether our predictive model can be generalized across Non-STEM courses.

REFERENCES